IN THE CLAIMS:

1. (Currently Amended) A method for generating a computer program, the method comprising:

receiving user input specifying a prototype, wherein the prototype comprises a series of functional operations, wherein at least one of the operations has an associated one or more parameters;

automatically generating a program that implements the prototype, in response to the specified prototype;

wherein said automatically generating the program comprises automatically generating a graphical user interface for the program;

wherein said automatically generating the graphical user interface comprises automatically creating graphical user interface elements [[controls]] associated with the one or more parameters, wherein during execution of the program, the graphical user interface elements are displayed and are operable to receive user input and/or display output.

2. (Cancelled)

3. (Original) The method of claim 1,

wherein said automatically generating the program comprises automatically generating code for the program without direct user input.

4. (Currently Amended) The method of claim 1,

wherein at least one of the operations has an associated input parameter;

wherein said generating the graphical user interface comprises creating a graphical user interface element [[control]] for interactively providing program input specifying a value for the input parameter.

5. (Currently Amended) The method of claim 1,

wherein at least one of the operations has an associated output parameter;

wherein said generating the graphical user interface comprises creating a graphical user interface element for viewing program output indicating a value for the output parameter.

6. (Currently Amended) The method of claim 1, wherein a plurality of parameters are associated with the functional operations, the method further comprising:

receiving user input specifying which of the plurality of parameters are desired to have associated graphical user interface elements [[controls]];

wherein said generating the graphical user interface comprises creating a graphical user interface element [[control]] associated with each specified parameter, but not creating graphical user interface elements [[controls]] associated with unspecified parameters.

7. (Original) The method of claim 1,

wherein the generated program is a text-based program.

8. (Currently Amended) The method of claim 1,

wherein the generated program is a graphical program, comprising a plurality of interconnected nodes that visually indicate functionality of the program.

9. (Original) The method of claim 1,

wherein said receiving user input specifying a prototype is performed by a prototyping application;

wherein the prototyping application interfaces with a programming environment application in order to perform said generating the program.

10. (Currently Amended) The method of claim 1, wherein at least one parameter has an associated data type, the method further comprising:

determining the data type of the at least one parameter;

wherein creating a graphical user interface element associated with the at least one parameter comprises creating a graphical user interface element according to the data type of the at least one parameter.

11. (Original) The method of claim 1,

wherein the prototype specifies an image processing algorithm;

wherein the generated program implements the image processing algorithm.

12. (Currently Amended) The method of claim 11,

wherein the generated program has a graphical user interface including one or more graphical user interface elements [[controls]] for providing input parameter values affecting the image processing algorithm.

13. (Currently Amended) The method of claim 11,

wherein the generated program has a graphical user interface including one or more graphical user interface elements [[controls]] for viewing output parameter values determined by the image processing algorithm.

14. (Currently Amended) A system for generating a computer program, the system comprising:

a prototyping environment application for receiving user input specifying a prototype, wherein the prototype comprises a series of functional operations, wherein at least one of the operations has an associated one or more parameters;

wherein the prototyping environment application is operable to automatically generate a program that implements the prototype, in response to the specified prototype;

wherein said automatically generating the program comprises automatically generating a graphical user interface for the program;

wherein said automatically generating the graphical user interface comprises automatically creating graphical user interface elements [[controls]] associated with the one or more parameters, wherein during execution of the program, the graphical user interface elements displayed and are operable to receive user input and/or display output.

15. (Cancelled)

16. (Original) The system of claim 14,

wherein said automatically generating the program comprises automatically generating code for the program without direct user input.

17. (Currently Amended) The system of claim 14,

wherein at least one of the operations has an associated input parameter;

wherein said generating the graphical user interface comprises creating a graphical user interface element for interactively providing program input specifying a value for the input parameter.

18. (Currently Amended) The system of claim 14,

wherein at least one of the operations has an associated output parameter;

wherein said generating the graphical user interface comprises creating a graphical user interface element for viewing program output indicating a value for the output parameter.

19. (Currently Amended) The system of claim 14,

wherein a plurality of parameters are associated with the functional operations;

wherein the prototyping environment application is operable to receive user input specifying which of the plurality of parameters are desired to have associated graphical user interface elements [[controls]];

wherein said generating the graphical user interface comprises creating a graphical user interface element associated with each specified parameter, but not creating graphical user interface elements [[controls]] associated with unspecified parameters.

20. (Original) The system of claim 14,

wherein the generated program is a text-based program.

21. (Currently Amended) The system of claim 14,

wherein the generated program is a graphical program, comprising a plurality of interconnected nodes that visually indicate functionality of the program.

22. (Original) The system of claim 14,

wherein the prototyping environment application interfaces with a programming environment application in order to perform said generating the program.

23. (Currently Amended) The system of claim 14,

wherein at least one parameter has an associated data type;

wherein the prototyping environment application is operable to determine the data type of the at least one parameter;

wherein creating a graphical user interface element associated with the at least one parameter comprises creating a graphical user interface element according to the data type of the at least one parameter.

24. (Original) The system of claim 14,

wherein the prototyping environment application is an image processing prototype environment application;

wherein the prototype specifies an image processing algorithm;

wherein the generated program implements the image processing algorithm.

25. (Currently Amended) The system of claim 24,

wherein the generated program has a graphical user interface including one or more graphical user interface elements [[controls]] for providing input parameter values affecting the image processing algorithm.

26. (Currently Amended) The system of claim 24,

wherein the generated program has a graphical user interface including one or more graphical user interface elements [[controls]] for viewing output parameter values determined by the image processing algorithm.

27. (Currently Amended) A memory medium comprising program instructions executable to:

receive user input specifying a prototype, wherein the prototype comprises a series of functional operations, wherein at least one of the operations has an associated one or more parameters;

automatically generate a program that implements the prototype, in response to the specified prototype;

wherein said automatically generating the program comprises automatically generating a graphical user interface for the program;

wherein said automatically generating the graphical user interface comprises automatically creating graphical user interface elements [[controls]] associated with the one or more parameters, wherein during execution of the program, the graphical user interface elements displayed and are operable to receive user input and/or display output.

28. (Cancelled)

29. (Original) The memory medium of claim 27,

wherein said automatically generating the program comprises automatically generating code for the program without direct user input.

30. (Currently Amended) The memory medium of claim 27,

wherein at least one of the operations has an associated input parameter;

wherein said generating the graphical user interface comprises creating a graphical user interface element for interactively providing program input specifying a value for the input parameter.

31. (Currently Amended) The memory medium of claim 27,

wherein at least one of the operations has an associated output parameter;

wherein said generating the graphical user interface comprises creating a graphical user interface element for viewing program output indicating a value for the output parameter.

32. (Currently Amended) The memory medium of claim 27, wherein a plurality of parameters are associated with the functional operations, wherein the program instructions are further executable to:

receive user input specifying which of the plurality of parameters are desired to have associated graphical user interface elements [[controls]];

wherein said generating the graphical user interface comprises creating a graphical user interface element associated with each specified parameter, but not creating graphical user interface elements [[controls]] associated with unspecified parameters.

33. (Original) The memory medium of claim 27,

wherein the generated program is a text-based program.

34. (Currently Amended) The memory medium of claim 27,

wherein the generated program is a graphical program, comprising a plurality of interconnected nodes that visually indicate functionality of the program.

35. (Original) The memory medium of claim 27,

wherein said receiving user input specifying a prototype is performed by a prototyping application;

wherein the prototyping application interfaces with a programming environment application in order to perform said generating the program.

36. (Currently Amended) The memory medium of claim 27, wherein at least one parameter has an associated data type, wherein the program instructions are further executable to determine the data type of the at least one parameter;

wherein creating a graphical user interface element associated with the at least one parameter comprises creating a graphical user interface element according to the data type of the at least one parameter.

37. (Original) The memory medium of claim 27,

wherein the prototype specifies an image processing algorithm;

wherein the generated program implements the image processing algorithm.

38. (Currently Amended) The memory medium of claim 37,

wherein the generated program has a graphical user interface including one or more graphical user interface elements [[controls]] for providing input parameter values affecting the image processing algorithm.

39. (Currently Amended) The memory medium of claim 37,

wherein the generated program has a graphical user interface including one or more graphical user interface elements [[controls]] for viewing output parameter values determined by the image processing algorithm.

40. (Currently Amended) A computer-implemented method for automatically generating a computer program, the method comprising:

receiving program information specifying functionality of the computer program;

automatically generating the computer program in response to the program information, wherein the computer program implements the specified functionality;

wherein said automatically generating the program comprises automatically generating a graphical user interface for the program;

wherein said automatically generating the graphical user interface comprises automatically creating one or more graphical user interface elements [[controls]] for providing input to and/or viewing output from the program, wherein during execution of the program, the graphical user interface elements displayed and are operable to receive user input and/or display output.

41. (Cancelled)

42. (Original) The method of claim 40,

wherein said automatically generating the computer program comprises automatically generating code for the program without direct user input.

43. (Currently Amended) The method of claim 40,

wherein each of the one or more automatically created graphical user interface elements [[controls]] corresponds to one or more parameters specified by the program information.

44. (Currently Amended) The method of claim 40,

wherein the generated computer program is a graphical program, comprising a plurality of interconnected nodes that visually indicate functionality of the program.

45. (Original) The method of claim 40,

wherein the received program information specifies one of:

a prototype;
a test executive sequence; and
a state diagram.

46. (New) The method of claim 1, wherein said automatically generating the program comprises:

automatically generating a block diagram, wherein the block diagram comprises a plurality of interconnected nodes that visually indicate the functionality of the program.

47. (New) The method of claim 1, wherein said generating the graphical user interface comprises:

automatically generating a user interface panel, wherein the user interface panel comprises the graphical user interface elements.